

CLAIMS

1. A method of lexicographically sorting data, comprising:
receiving a set of N cyclic shifts of N characters identifiable by an array of indexes
5 {0, 1, 2, ..., N-1};
sorting the set of cyclic shifts based on a comparison of a first character of each cyclic shift;
for an nth sorting iteration of the set of cyclic shifts, where n = 1, 2, 3, ..., up to $2^n > N$:
10 sorting at least a subset of the cyclic shifts which are identifiable by a subset array of indexes in the array in accordance with a previous sort of cyclic shifts associated with the subset array of indexes plus $2^{(n-1)} * \text{modulo}(N)$; and
repeating the sorting for a next nth sorting iteration as necessary until the set of cyclic shifts are lexicographically sorted.
15
2. The method of claim 1, further comprising:
identifying the subset of the cyclic shifts with equivalent $2^{(n-1)}$ characters for the nth sorting iteration.
20
3. The method of claim 1 wherein after the nth sorting iteration, $2^{(n-1)}$ leading characters in the cyclic shifts have been sorted.
25
4. The method of claim 1, wherein the act of sorting based on the comparison of the first character comprises a radix sort.
5. The method of claim 1, wherein the method is included in a data compression algorithm.
30
6. The method of claim 1, wherein the method is included in a Burrows-Wheeler Transform (BWT) clustering procedure.

7 A computer program product, comprising:
memory;
computer instructions stored in the memory; and
5 the computer instructions being executable by a processor for lexicographically sorting cyclic data by:

receiving a set of N cyclic shifts of N characters identifiable by an array of indexes {0, 1, 2, ..., N-1};

10 sorting the set of cyclic shifts based on a comparison of a first character of each cyclic shift;

 for an nth sorting iteration of the set of cyclic shifts, where n = 1, 2, 3, ..., up to $2^n > N$:

15 sorting at least a subset of the cyclic shifts which are identifiable by a subset array of indexes in the array in accordance with a previous sort of cyclic shifts associated with the subset array of indexes plus $2^{(n-1)} * \text{modulo}(N)$;
 and

 repeating the sorting for a next nth sorting iteration as necessary until the set of cyclic shifts are lexicographically sorted.

20 8. The computer program product of claim 7, wherein the computer instructions are further executable for:

 identifying the subset of the cyclic shifts with equivalent $2^{(n-1)}$ characters for the nth sorting iteration.

25 9. The computer program product of claim 7 wherein after the nth sorting iteration by the computer instructions, $2^{(n-1)}$ leading characters in the cyclic shifts have been sorted.

30 10. The computer program product of claim 7, wherein the act of sorting based on the comparison of the first character comprises a radix sort.

11. The computer program product of claim 7, wherein the computer instructions are included in a data compression algorithm.

12. The computer program product of claim 7, wherein the computer instructions are included in a Burrows-Wheeler Transform (BWT) clustering procedure.

5 13. The computer program product of claim 7, wherein a computer embodies the computer program product.

10 14. The computer program product of claim 7, wherein a mobile communication device embodies the computer program product.

15. A system for communicating data, comprising:
a wireless packet data network;
a mobile communication device which operates in the wireless packet data network;
a computer coupled to the wireless packet data network;
the computer including a computer readable memory;
computer instructions stored in the computer readable memory;
the computer instructions being executable by a processor for performing a data compression algorithm to generate compressed data which is communicated to the mobile communication device, the data compression algorithm including a sorting algorithm for lexicographically sorting cyclic data by:

20 receiving a set of N cyclic shifts of N characters identifiable by an array of indexes {0,1, 2, ..., N-1};

25 sorting the set of cyclic shifts based on a comparison of a first character of each cyclic shift;

for an nth sorting iteration of the set of cyclic shifts, where n = 1, 2, 3, ..., up to $2^n > N$:

30 sorting at least a subset of the cyclic shifts which are identifiable by a subset array of indexes in the array in accordance with a previous sort of

cyclic shifts associated with the subset array of indexes plus $2^{(n-1)*\text{modulo}(N)}$;
and

repeating the sorting for a next nth sorting iteration as necessary until
the set of cyclic shifts are lexicographically sorted.

5

16. The system of claim 15, wherein the computer instructions are further
executable for:

identifying the subset of the cyclic shifts with equivalent $2^{(n-1)}$ characters for the nth
sorting iteration.

10

17. The system of claim 15 wherein after the nth sorting iteration by the computer
instructions, $2^{(n-1)}$ leading characters in the cyclic shifts have been sorted.

15

18. The system of claim 15, wherein the act of sorting based on the comparison of
the first character comprises a radix sort.

19. The system of claim 15, wherein the computer instructions are included in a
data compression algorithm.

20

20. The system of claim 15, wherein the computer instructions are included in a
Burrows-Wheeler Transform (BWT) clustering procedure.

25

21. A method of lexicographically sorting data, comprising:
receiving a plurality of N cyclic shifts of N characters to be sorted;
providing a sorting array of N elements $\{0, 1, 2, \dots, N-1\}$, each element being an
index which uniquely corresponds to one cyclic shift of the plurality of N cyclic shifts;
initially sorting the plurality of N cyclic shifts of N characters based on a first
character of each cyclic shift of N characters, and ordering the indexes in the sorting array in
accordance with such sorting;

30

iteratively sorting the initially sorted plurality of N cyclic shifts by:

for an n th sorting iteration of the set of cyclic shifts, where $n = 1, 2, 3, \dots$, up to $2^n > N$:

sorting at least a subset of the cyclic shifts which are identifiable by a subset array of indexes in the sorting array by sorting the subset array of indexes in accordance with a previous sort of cyclic shifts for the subset array of indexes plus $2^{(n-1)} * \text{modulo}(N)$; and

repeating the sorting for a next n th sorting iteration as necessary until the set of cyclic shifts are lexicographically sorted.

10 22. The method of claim 21, further comprising:

providing an inverse sorting array of N elements, each element identifying a sorted position for the cyclic shift corresponding to a position of the element in the inverse sorting array; and

utilizing the inverse sorting array for sorting the subset array of indexes.

15

23. The method of claim 21, further comprising:

providing an equivalents array of N elements for identifying the subset array of indexes corresponding to cyclic shifts having equivalent characters.

20

24. The method of claim 21, further comprising:

providing an inverse sorting array of N elements $\{0, 1, 2, \dots, N-1\}$, each element identifying a sorted position for the cyclic shift corresponding to the position of the element in the inverse sorting array;

providing an equivalents array of N elements for identifying the subset array of indexes corresponding to cyclic shifts having equivalent characters; and

utilizing the inverse sorting array for sorting the subset array of indexes.

25. The method of claim 21 wherein after the nth sorting iteration, $2^{(n-1)}$ leading characters in the cyclic shifts have been sorted.

30